//create a model

var Todo = Backbone.Model.extend({

defaults: function(){

return {

title: 'default-to-do-list',

taskDone: false

};

},

initialize:function(){

},

toggle: function(){

this.save({ taskDone: !this.get('taskDone')});

}

});

//create a COLLECTION

var ToDoCollection = Backbone.Collection.extend({

model: Todo,

localStorage: new Backbone.LocalStorage("backbone-todo-storage"),

completed:function(){

return this.where({completed:true}) ;

},

taskDone:function(){

return this.filter(function(todo){return todo.get('taskDone');});

}

});

// instance of the Collection

var toDoCollection = new ToDoCollection();

//create a view

var TodoView = Backbone.View.extend({

//tagName: 'li',

events: {

'dblclick label' : 'edit',

'keypress .edit' : 'addTask',

'blur .edit' : 'close',

'click .toggle': 'toggleDone',

'click .destroy': 'destroy'

},

initialize: function(){

this.template =\_.template($('#template1').html());

//listen to change in the model

this.listenTo(this.model, 'change', this.render);

//removing the view

this.listenTo(this.model, 'destroy', this.remove);

},

render: function(){

//toJSON: is used to clone a model's attributes into plain object to be used in template rendering

this.$el.html(this.template(this.model.toJSON()));

// template: \_.template($('#template1')),

this.$el.toggleClass('taskDone', this.model.get('taskDone'));

this.input = this.$('.edit');

return this;

},

//edit the input

edit: function(){

//this.$el.addClass('editing');

this.input.removeClass('hide');

//this.input.focus();

},

//save changes and close the edit mode

close: function(){

var value = this.input.val();

if(value) {

this.model.save({title: value});

}

// this.$el.removeClass('editing');

},

//add a new task by hitting enter -(save the current one and be able to add another)

addTask: function(event){

if(event.which == 13){

this.close();

}

},

//toggle the state of task

toggleDone: function(){

this.model.toggle();

},

//delete the model by removing the item from the list

destroy: function(){

this.model.destroy();

}

});

//viewing the entire list

var AppView = Backbone.View.extend({

events: {

'keypress #new-todo': 'AddNewEntry',

'click #destroyAll' : 'DeleteAllItems'

},

initialize: function () {

this.template =\_.template($('#template2').html());

this.input = this.$('#new-todo');

this.footer = this.$('footer');

this.listenTo(toDoCollection, 'add', this.addSingleItem);

this.listenTo(toDoCollection, 'reset', this.addAllItems);

this.footer =this.$('footer');

toDoCollection.fetch(); /\* Loads the entire list from local storage - u want your

to-do items to stay even after refresh or closing and opening the browser again \*/

/\* toDoCollection.fetch();\*/

},

render: function()

{

var taskDone = toDoCollection.taskDone().length;

if(toDoCollection.length)

{

this.footer.show();

this.footer.html(this.template2({taskDone: taskDone}));

}

else

{

this.footer.hide();

}

},

AddNewEntry: function(event){

if ( event.which !== 13 || !this.input.val() ) {

return;

}

toDoCollection.create({title: this.input.val(), taskDone: false});

this.input.val(''); // clean input box

},

addSingleItem: function(todoitem){

var view = new TodoView({model: todoitem});

this.$('#todo-list').append(view.render().el);

},

DeleteAllItems: function(){

\_.invoke(toDoCollection.taskDone(),'destroy');

return false;

},

addAllItems: function(){

this.$('#todo-collection').html(''); // clean the todo list

toDoCollection.each(this.addSingleItem, this);

}

});

//create the app (or initialize the view)

$(document).ready(function(){

var appView = new AppView({

el: $('#todoapp')

});

});